Available online at www.sciencedirect.com

ScienceDirect

Procedia Computer Science 00 (2023) 000–000

Procedia Computer Science

www.elsevier.com/locate/procedia

27th International Conference on Knowledge-Based and Intelligent Information & Engineering Systems (KES 2023)

Knowledge Organization System for Partial Automation to Improve the Security Posture of IoMT Networks

Kulsoom S. Bughio[a], Leslie F. Sikos[a], Helge Janicke[a]

[a]*Edith Cowan University, Joondalup 6027, Western Australia, Australia*

Abstract

Remote patient monitoring is a healthcare delivery model that uses technology to collect and transmit patient data from a remote location to healthcare providers for analysis and treatment. Remote patient monitoring systems rely on a network infrastructure to gather and transmit data from patients to healthcare providers through a network. While these systems become more prevalent, they may also become targets for cyberattacks. This paper deals with the development of a domain ontology to facilitate partial automation in improving the security posture of IoT networks used in remote patient monitoring. For this purpose, it captures the semantics of the concepts and properties of the main security aspects of IoT medical devices. This will be complemented by a comprehensive ruleset, SPARQL queries, and a mechanism to enable automated reasoning over the aggregated knowledge.

Keywords: Internet of Medical Things; Remote Patient Monitoring; Medical Devices; Knowledge Organization System; SWRL Rules; Automated Reasoning; SPARQL

1. Introduction

Over the last two decades, advancements in information and communication technology have enabled the design and development of real-time healthcare solutions. The COVID-19 pandemic has accelerated the implementation of such systems, particularly for those living with chronic medical conditions, including but not limited to, diabetes, cardiovascular disease etc. In this regard, the Internet of Medical Things (IoMT) has played a pivotal role to connect patients with doctors and hospitals. IoMT is the collection of IoT technologies, which monitors a patient's health and transmits the patient's information through a network to the server i.e. cloud server [1]. Remote Patient Monitoring (RPM) is one of the applications in the IoMT, where a continuous stream of real-time information about patients can be transferred through medical devices connected with IoT sensors. This allows healthcare providers to monitor patients' health status and manage their care remotely, without the need for in-person visits [2].

On the one hand, where IoMT in remote patient monitoring provide the facility to patients, at the same time, it faced many challenges in terms of heterogeneity and interoperability. Knowledge organization systems (KOS) are one of the solutions to support and organize the data coming from different sources. A KOS is a systematic and organized way for information retrieval, access, and use of knowledge resources. KOSs include various types of taxonomies, ontologies, thesauri, and other controlled vocabularies [3]. KOSs can help identify and manage patient data, and provide standardized terminology by capturing the semantics of their medical devices and properties. In such a way, semantic models are used to represent medical information and knowledge in a structured and machine-readable format which facilitates the sharing and integration of information among various systems such as cybersecurity. For example, an ontology-based KOS can improve the interoperability of different IoMT devices, allowing for efficient and standardized communication between devices and systems. Additionally, it can help identify security risks and support security measures by standardizing the vocabulary and terms used to describe security issues.

In this article, we proposed an ontology called IoMT ontology for the semantic representation of Medical IoT devices. SWRL rules deduce the security aspects of IoT medical devices. The rest of the article is organized as follows: Section 2 describes the related work for knowledge organization systems. Section 3 provides the propped ontology for our work. The conclusion is provided in the last section 4.

2. Related Work

Many researchers work with knowledge organization systems where they deal with medical devices, various sensors, standards etc., to provide benefits to patients, doctors, and healthcare providers. To promote interoperability in the healthcare sector, authors in [4], introduced an IoT-based system for collecting, integrating, and interconnecting IoT data in medical emergency services. For this purpose, they proposed a semantic data model that can uniformly store and interpret diverse IoT data, which is particularly advantageous in real-time applications. This resource-based method for accessing IoT data is effective in distributed heterogeneous environments, enabling prompt access to data on mobile and cloud computing platforms. To tackle the challenge of device heterogeneity, the Researchers [5] developed a mechanism by integrating and understanding the application programming interfaces (APIs) of different medical devices to collect their data. They proposed a method for constructing ontologies of the various APIs from both known and unknown medical devices and identifying their syntactic and semantic similarity with the help of mapping techniques. A semantic model was proposed by researchers [6] based on an IoT platform named SM-IoT. This model facilitates interoperability between medical devices and data sources and includes contract-based security policies to ensure the confidentiality of patient health data. In our research, an ontology is developed based on expert knowledge in the domain of remote patient monitoring systems. This ontology detects compromised medical devices due to vulnerabilities and performs automated reasoning.

3. Proposed MIoT Ontology

In this research, the authors are inspired by Knowledge Engineering Methodology used in Ontology Development 101 [7] and develop the concepts and relationships from two different areas i.e., Remote Patient Monitoring (RPM) and Cybersecurity. By applying ontology engineering practices, the developed ontology is named as MIoTOntology; it stores the knowledge related to the domain and information about vulnerabilities in medical IoT devices. This ontology defines Specification and Scope, and Conceptualization as a foundation for this research, where knowledge area, the purpose of research; concepts, relationships and terms used in this research. The implementation and evaluation are provided as the final stage of the proposed ontology.

3.1. Specification and Scope

This phase involves defining the scope of the ontology, identifying the relevant concepts and relationships, and determining the purpose of the ontology. It provides a foundation for the subsequent phase of ontology development, which is conceptualization. The healthcare domain is very sensitive to cyberattacks and vulnerabilities due to the exposure of heterogeneous data on the web. The knowledge organization systems are one of the solutions which support cyber-resilience in these systems by defining the semantics of those systems. The MIoT ontology focused on

those participants who are relevant to a remote patient monitoring system such as Patients, Doctors, Nurses, and caregivers. This ontology shows how to detect vulnerabilities in remote patient monitoring systems by using semantic standards and technologies. Furthermore, the relationship between entities is defined in domain ontology, is based on various sources and tackles the information in terms of medical devices, their connectivity, and their vulnerabilities.

3.2. Conceptualization

This phase involves converting the foundation into formalization and defining concepts, roles and individuals and their relationships in a machine-interpretable. The axioms are created that specify the relationships between the concepts. Here are some examples of concepts, roles, individuals, and axioms are given below.

- *Concepts (Classes):*

  Vendor, Product, MedicalDevice, MedicalSensor, Service, Technology, Protocol, Standard, Person, Doctor, Patient and more. For example, Axioms are provided for a Concept Product and its subclass MedicalDevice;

  :Product a owl:Class .
  :MedicalDevice a owl:Class ; rdfs:subClassOf :Product .

- *Roles (properties)*

  Object Properties: connectedToNetwork, exploitedBy, hasAffectOn, hasDeviceType and more.
  Datatype properties: hasDosageValue, hasModifiedData, hasReadingValue and more.
  Datatypes: String, Integer, Decimal and Boolean

- *Individuals (Instances)*

  AccuCheck, BloodPressure, BloodPressureMonitor, BloodPressureReading1, BloodPressureReading2, BloodPressureReading3 HeartRateMonitor, Scale, Thermometer and more.

3.3. Implementation

This ontology is implemented in Protégé[1] Version 5.6.1 and populated with a few instances. The Protégé is an open-source software tool used for building knowledge-based systems, including ontologies. The MIoT ontology hierarchical Classes and subclasses are shown in Figure 1. The ontology used the top-down approach where classes are divided into subclasses hierarchically shown in Fig. 1., and its alpha version is available at https://archive.org/download/MIoTOntology.

3.3.1 Inference Process

In this phase, some rules are defined to achieve the objectives of our research. Such as., (a) inferring the reading of the blood pressure monitor if it's different from usual, (b) inferring the value of the medical device if it has modified data for medication dosage (c) generating the adverse effects if medication dosage is modified, and (d) recommending the manual reading for the blood pressure monitor if it is compromised alert the patient if medication dosage is modified it leads to adverse effects. The inference rules are specified with the SWRL (Semantic Web Rule Language)[2] language using the SWRLTab plug-in. These SWRL rules can be used to demonstrate and infer new data in the below assumptions where *John* is an *instance* for the class *Patient,* and he is using the *blood pressure monitor (instance* of

[1] https://protege.stanford.edu/
[2] https://www.w3.org/Submission/SWRL/

Medical Device*)* and *medication dosage* as Medical Function.

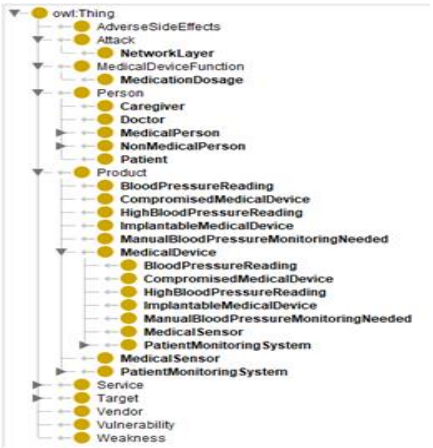

Fig. 1. The hierarchical concepts for MIoT ontology

- If the system is vulnerable and the reading of the blood pressure monitor is different from usual. The rule will trigger and check, if the blood pressure reading is greater than 120 (120/80 is normal blood pressure reading) it infers that the reading is HighBloodPressureReading.

```
BloodPressureReading(?r),hasReadingValue(?r,?reading),
greaterThan(?reading, 120) ->
HighBloodPressureReading(?r)                    Rule ------>1
```

This rule satisfies the condition in (a). This rule can be used to detect a blood pressure reading when a patient's blood pressure readings are higher than normal.

- If a medical device such as medication dosage is connected to the network and if its data is modified, it infers that the device is a CompromisedMedicalDevice.

```
MedicalDevice(?d),connectedToNetwork(?d,?networklayer),
hasModifiedData(?networklayer,True)-> CompromisedMedicalDevice(?d)
                              Rule ------>2
```

This rule can be used to detect when a medical device has been compromised for (b).

- The adverse side effects are detected by checking, if a MedicationDosage has a dosage value greater than 10, (The dosage value is between 5-8 in the system) it will infer that *AdverseSideEffects* may occur for the patient which leads to life-threatening.

```
MedicationDosage(?d),   hasPatient(?d,   ?p),   hasDosageValue(?d,
?dosage), greaterThan(?dosage, "10") ->
AdverseSideEffects(?p)                          Rule ------>3
```

This rule can be used for (c) to detect when a patient is experiencing adverse side effects due to altered medication dosages.

- If a medical device such as a blood pressure monitor is CompromisedMedicalDevice of type "Blood Pressure Monitor" is detected. It infers that the *ManualBloodPressureMonitoringNeeded* is recommended for the patients until the issue is resolved.

```
CompromisedMedicalDevice(?d),
hasDeviceType(?p,?bloodPressureMonitor)->
ManualBloodPressureMonitoringNeeded(?p)                Rule ------>4
```

This rule can be used to recommend manual blood pressure monitoring for a patient whose blood pressure monitor has been compromised as discussed in (d).

The results depicted in the following Fig. 2. show the inferred axioms in OWL format after running the Drools engine.

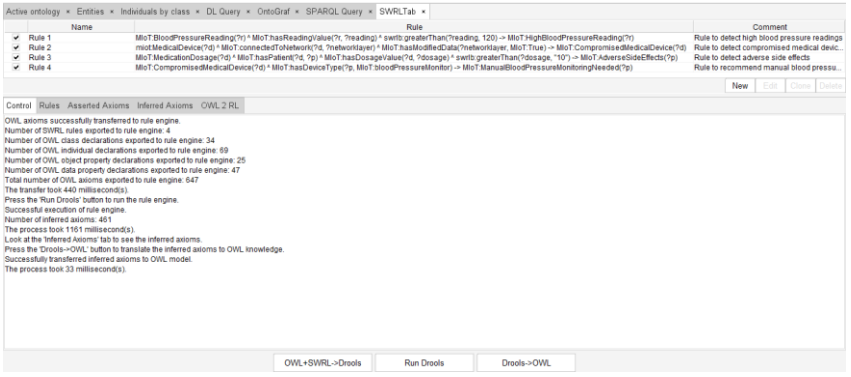

Fig. 2. Running Drools Inference Engine

3.4. Evaluation

There are many reasoning engines available for ontology models such as Jena[3], Fact++[4], Racer[5], Hermit[6] and Pellet[7] and so on. In this research, the Pellet reasoner is used to evaluate the consistency of the ontology. Pellet is an OWL-DL reasoner that offers extensive middleware and several unique features while achieving acceptable to very good performance. Additionally, it implements several OWL-DL extensions, such as a combination formalism for OWL-DL ontologies, and preliminary support for OWL/Rule hybrid reasoning [8]. The result generated by the Pellet reasoner is shown in Fig. 3.

The Pellet reasoner uses the tableau algorithm which is an effective and widely used algorithm for OWL-DL reasoning. The tableau algorithm is a backward chaining algorithm, meaning that it starts with a goal or query and works backwards to find the necessary conditions to satisfy that goal. In the context of OWL-DL reasoning, the algorithm starts with the negation of the class expression to be checked for satisfiability and tries to find a contradiction. If a contradiction is found, the original class expression is unsatisfiable. Otherwise, the algorithm tries

[3] https://jena.apache.org/documentation/inference/
[4] http://owl.cs.manchester.ac.uk/tools/fact/
[5] https://www.ifis.uni-luebeck.de/~moeller/racer/
[6] http://www.hermit-reasoner.com/
[7] https://www.w3.org/2001/sw/wiki/Pellet

to apply tableau expansion rules to derive new subexpressions, which are added to the tableau.

```
INFO  16:49:56  ------------------------------ Running Reasoner ------------------------------
INFO  16:49:56  Pre-computing inferences:
INFO  16:49:56      - class hierarchy
INFO  16:49:56      - object property hierarchy
INFO  16:49:56      - data property hierarchy
INFO  16:49:56      - class assertions
INFO  16:49:56      - object property assertions
INFO  16:49:56      - data property assertions
INFO  16:49:56      - same individuals
INFO  16:49:56  Ontologies processed in 190 ms by Pellet
INFO  16:49:56
```

Fig. 3. Reasoning in Pellet Reasoner

After validation of ontology, it can be queried by users to deduce the implicit facts. The SPARQL query language allows extracting the knowledge from MIoTontology. Here we present some queries along with their results and description.

1.
```
SELECT ?reading
WHERE {
?reading rdf:type MIoT:BloodPressureReading .
?reading MIoT:hasReadingValue ?value.
 FILTER (?value > 120)
}
```

reading
BloodPressureReading1
BloodPressureReading2

Fig. 4. SPARQL Query result

The BloodPressureReading class has three instances. This query selects all instances of the `BloodPressureReading` class and their associated `hasReadingValue` properties and applies a filter that checks if the reading value is greater than 120 as shown in Fig. 4.. The query returns a list of `?reading` variables that correspond to blood pressure readings with values greater than 120.

2.
```
SELECT ?reading
WHERE {
?reading rdf:type MIoT:BloodPressureReading .
?reading MIoT:hasReadingValue ?value .
FILTER (?value < 120)
}
```

reading
BloodPressureReading3

Fig. 5. SPARQL Query result

This query selects all instances of the `BloodPressureReading` class and their associated `hasReadingValue` properties and applies a filter that checks if the reading value is greater than 120. The query returns a list of `?reading` variables that correspond to blood pressure readings with values lesser than 120 as shown in Fig. 5.

```
3. SELECT ?value
   WHERE {
   ?bloodPressureReading1 rdf:type MIoT:BloodPressureReading .
   ?bloodPressureReading1 MIoT:hasReadingValue ?value .
   ?bloodPressureReading1 MIoT:hasPatient MIoT:John .
   }
```

value
"140"^^<http://www.w3.org/2001/XMLSchema#integer>

Fig. 6. SPARQL Query result

This query selects all instances of the `BloodPressureReading` class and their associated `hasReadingValue`, and `hasPatient`. It applies a filter that targets readings for the specific patient "John" (`?reading MIoT:hasPatient MIoT:John`), as shown in Fig. 6.

4. Conclusion

Remote Patient Monitoring involves IoMT to gathering patient data from a remote location and transmitting it to healthcare providers for analysis and treatment Knowledge organization systems are the best solutions to organize and arrange that information systematically. In this regard, MIoTOntology captures the semantics of the concepts and properties of the primary security aspects of IoT medical devices used in remote patient monitoring systems. SWRL rules are used to infer new facts and provided automated reasoning to enhance the security of IoMT networks used in Remote Patient Monitoring. Finally, SPARQL is used to extract the knowledge from the proposed ontology.

References

[1] M. Wazid, A. K. Das, J. J. P. C. Rodrigues, S. Shetty, and Y. Park, 'IoMT Malware Detection Approaches: Analysis and Research Challenges', *IEEE Access*, vol. 7, pp. 182459–182476, 2019, doi: 10.1109/ACCESS.2019.2960412.

[2] K. S. Bughio and L. F. Sikos, *Knowledge Organization Systems to Support Cyber-Resilience in Medical Smart Home Environments*. Springer International Publishing, 2023. doi: 10.1007/978-3-031-24946-4_5.

[3] T. Baker, S. Bechhofer, A. Isaac, A. Miles, G. Schreiber, and E. Summers, 'Key choices in the design of Simple Knowledge Organization System (SKOS)', *Journal of Web Semantics*, vol. 20, pp. 35–49, 2013, doi: 10.1016/j.websem.2013.05.001.

[4] B. Xu, L. Da Xu, H. Cai, C. Xie, J. Hu, and F. Bu, 'Ubiquitous data accessing method in iot-based information system for emergency medical services', *IEEE Trans Industr Inform*, vol. 10, no. 2, pp. 1578–1586, 2014, doi: 10.1109/TII.2014.2306382.

[5] A. Mavrogiorgou, A. Kiourtis, and D. Kyriazis, 'Identification of IoT medical devices APIs through ontology mapping techniques', *EAI/Springer Innovations in Communication and Computing*, pp. 39–54, 2020, doi: 10.1007/978-3-030-30335-8_4.

[6] A. Dridi, S. Sassi, and S. Faiz, 'Towards a semantic medical internet of things', *Proceedings of IEEE/ACS International Conference on Computer Systems and Applications, AICCSA*, vol. 2017-Octob, pp. 1421–1428, 2018, doi: 10.1109/AICCSA.2017.194.

[7] N. F. Noy and D. L. McGuinness, 'Ontology Development 101: A Guide to Creating Your First Ontology', *Stanford Knowledge Systems Laboratory*, p. 25, 2001, doi: 10.1016/j.artmed.2004.01.014.

[8] E. Sirin, B. Parsia, B. C. Grau, A. Kalyanpur, and Y. Katz, 'Pellet: A practical OWL-DL reasoner', *Journal of Web Semantics*, vol. 5, no. 2, pp. 51–53, Jun. 2007, doi: 10.1016/J.WEBSEM.2007.03.004.